Action Recognition from Weak Alignment of Body Parts – Supplementary Material

Minh Hoai[1]
http://www.robots.ox.ac.uk/~minhhoai/
L'ubor Ladický[2]
http://www.inf.ethz.ch/personal/ladickyl/
Andrew Zisserman[1]
http://www.robots.ox.ac.uk/~az/

[1] Oxford University, UK
[2] ETH Zürich, Switzerland

1 Details of Feature Extraction

We train a kernel SVM for each action class. The SVM kernel is a convex combination of base kernels, which capture different visual cues: HOG, SIFT, color, pose, object detection scores. Some of these cues are computed at various relative locations of the provided human bounding box, yielding a total of 20 kernels. We optimize the weights for kernel combination using randomized grid search. This section describes these kernels and implementation details.

1.1 HOG-based kernels

We compute kernels at several relative locations of the provided person bounding box. In particular, we compute the parts-aligned kernels for *bbox*, *up-bbox*, *low-bbox*, *ex-bbox*, as illustrated in Fig. 1. We additionally compute the default kernel at the detected upper body. This yields a total of 5 HOG-based kernels. Here, an image region, given by a rectangular bounding box, is represented by HOG descriptors [2] computed at the locations of its parts. We use the 3-level part model as illustrated in Figure 2 of the main paper. First, each part is normalized to a standard size of 24×24 pixels. This normalized patch is divided into 9 8×8-pixel cells, and a 9×32-dim HOG descriptor is computed for the patch (using DPM implementation [3]). Finally, the HOG descriptors of all parts are concatenated to create a single feature vector of 6048 (=21×9×32) dimensions.

1.2 SIFT-based kernels

Using SIFT descriptor [7], we compute a default kernel for each of the following locations: *bbox*, *ub-bbox*, *low-bbox*, *ex-bbox*, *lrb-bbox*, *lr-bbox*, *bckgrnd* (see Fig. 1). We also compute a kernel for the detected upper body. Additionally, we compute a second kernel for *bbox*, discarding SIFT features outside the silhouette. This generates a total of 9 SIFT-based kernels. Here, an image region is represented by the spatial pyramid bag-of-words representation of SIFT descriptors [7]. The rest of this subsection describes details about SIFT and codebook generation.

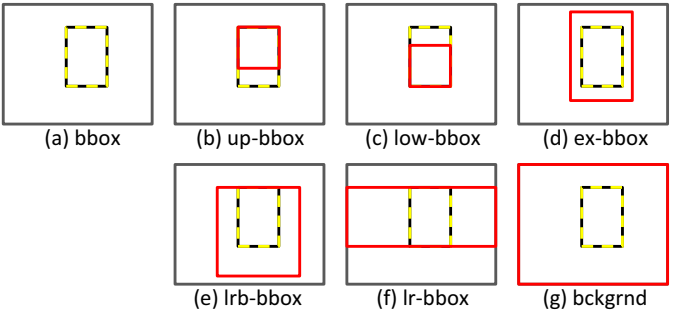

Figure 1: **Relative locations for the region of interest**. In each subfigure, the relative location is delineated by the solid red box, while the person bounding box is the black-and-gold rectangle.

For each image in our dataset, with the provided person bounding box, we first resize the image so that the larger dimension of the resized person bounding box is 200 pixels. We compute dense SIFT [7] (using VLFEAT [8]) descriptors, and following the suggestion of [1], each 128-dim SIFT descriptor is L_1-normalized.

For an expressive codebook with small size, we learn a codebook with product of quantization [4]. We divide each 128-dim SIFT vector into two blocks of 64 dimensions. For each block, we use k-means to learn a codebook of 256 visual words. Descriptors from both training and test sets are assigned to one of 256 visual words (one for each block) and aggregated into (2×256)-dimensional histogram. Following the spatial pyramid representation [5], we use 3-level part model as shown in Fig. **??**. Local histograms within each part are concatenated into one feature vector. This yields a 10752 (=512×(1+4+16)) dimensional vector, which is subsequently L_1-normalized.

1.3 Color-based kernel

The construction of color-based kernels is similar to that of HOG-based kernels. The main difference is to replace the HOG descriptor by a color descriptor. Here, our color descriptor is the histogram of *normalized-rg* values. Normalized-rg values are computed as: $nr = r/(r+g+b), nb = b/(r+g+b)$ for an (r,g,b) pixel, and we use a joint histogram of 256 bins. We compute a parts-aligned kernel for the provided bounding box and another parts-aligned kernel for the same area, but discarding pixels outside the silhouette. Finally, we average these two kernels to produce a single color-based kernel.

1.4 Pose-based kernels

We use the silhouette and the upper body as a proxy for the pose and construct two kernels. One of them is similar to HOG-based kernels, except we use an occupancy descriptor instead of the HOG descriptor. The occupancy descriptor divides a part (of the 3-level part model) into four quarters. It represents the part by a 5-dimensional vector, encoding the proportion of silhouette pixels inside the part and inside each quarter. The first pose-based kernel is computed for this occupancy descriptor at the provided bounding box. The second pose-based kernel is computed based on the relative location of the detected upper body w.r.t. the bounding box and the relative location of the bounding box w.r.t. the entire image.

	color		silhouette		SIFT		SIFT+silhouette	
	DF	PA	DF	PA	DF	PA	DF	PA
jumping	40.6	40.3	40.7	**46.5**	53.9	**55.2**	51.2	**53.9**
phoning	14.6	14.8	13.4	**15.7**	30.5	31.3	32.5	31.9
play'instru	31.1	**37.5**	13.1	12.7	53.2	**54.5**	47.7	**48.9**
reading	15.2	16.1	17.8	**19.1**	30.3	30.2	31.8	31.7
ridingbike	36.2	**39.9**	20.0	**24.6**	67.2	**68.2**	64.0	**65.0**
ride'horse	35.8	**41.5**	33.0	**42.0**	75.5	**76.7**	68.4	**69.4**
running	42.9	**44.3**	48.1	**49.5**	60.8	**61.8**	60.7	**61.7**
take'photo	10.6	**11.7**	16.2	15.9	23.8	23.3	22.1	21.5
usingcomp	17.3	**18.8**	21.2	21.2	45.7	**47.3**	37.7	**38.8**
walking	28.4	**32.0**	35.2	**40.8**	57.1	56.8	52.1	52.2
mean	27.3	**29.7**	25.9	**28.8**	49.8	**50.5**	46.8	**47.5**

Table 1: **Average precision for different feature types**. *color*: based on the histogram of normalized-rg values. *silhouette*: based on the occupancy of silhouette pixels. *SIFT*: BoW representation. *SIFT+silhouette*: BoW representation, but discarding SIFT descriptors computed outside the silhouette. For each double column, the bigger values that are at least 1% better than corresponding values are printed in bold.

1.5 Object-based kernels

Following [6], which suggest using object-level features for image classification, we investigate the use of object detection scores for human action recognition. This subsection describes how to compute a feature vector from the outputs of DPM detectors. We also experimented with Object-bank [6], but this did not improve the performance. We omit the experiments using Object-bank in this paper.

We compute the feature vector using a collection of object detectors as follows. We first resize the image so that the larger dimension of bounding box of the person performing the action is 300 pixels. Centering the image on the bounding box, we cropped the image so that its width and height are 1.5 the width and height of the bounding box. We run each object detector on the cropped image and record the highest detection score. Finally, we concatenate these detection scores to create a feature vector (the dimension of the feature vector is the number of object detectors).

We utilize three sets of object detectors. The first set contains 20 detectors which come with the DPM implementation [3]. These detectors were trained on VOC2009 data. For the second set of detectors, we train one DPM for each action class of interest in a one-vs-all fashion; these detectors were trained on the VOC2012 training data. The third set contains 16 musical instrument detectors, which were trained on ImageNet. We treat three sets of detectors separately, yielding three separate RBF kernels.

2 Additional experiment results

Tab. 1 is complementary to the Table 2 of the main paper. It reports the detailed performance, APs for individual classes. for several types of visual descriptors. In all cases, PA significantly outperforms DF on many action classes.

	up-bbox		low-bbox		ex-bbox		combine	
	DF	PA	DF	PA	DF	PA	DF	PA
jumping	52.4	**57.1**	36.3	**40.3**	49.5	**50.8**	60.6	**63.5**
phoning	24.8	**26.9**	21.7	**24.0**	25.8	26.6	28.7	**30.5**
play'instru	44.8	**46.6**	33.6	**37.9**	43.0	**46.5**	56.2	**57.4**
reading	26.3	**27.4**	22.7	**25.0**	19.9	**22.1**	29.4	**30.6**
ridingbike	53.5	**56.7**	70.6	**72.4**	73.3	**74.4**	83.1	84.0
ridinghorse	39.2	**40.7**	66.0	**68.0**	73.2	**75.8**	80.0	79.7
running	46.0	**48.3**	57.8	**61.0**	61.1	61.3	69.7	**72.9**
takingphoto	15.9	**18.1**	15.3	**16.9**	19.0	18.4	22.5	23.1
usingcomp	35.5	**37.5**	39.6	40.5	39.0	**43.0**	46.0	**50.4**
walking	48.8	48.7	47.9	**50.2**	52.7	**53.8**	65.6	65.7
mean	38.7	**40.8**	41.2	**43.6**	45.7	**47.3**	54.2	**55.8**

Table 2: **Experiments on different locations of the person bounding box** – comparison of part-aligned kernels and the default kernels using HOG descriptors. *ex-bbox*, *up-bbox*, *low-bbox* are defined in Fig. 1. *combine*: using the average of the kernels computed for *bbox*, *ex-bbox*, *up-bbox*, *low-bbox*. For each double column, the bigger values that are at least 1% better than corresponding values are printed in bold.

Tab. 2 displays results for HOG-based kernels computed at different relative locations of the provided bounding box (*bbox*). *up-bbox* and *low-bbox* are obtained by restricting the *bbox* to its upper or lower portions, while *ex-bbox* is the extended *bbox*, as shown in Fig. 1. In all cases, PA has significant advantage over DF. Compared with the results on the *bbox* given in Table 1 of the main paper, the relative advantage of PA is more significant. This suggests bigger benefits for loosely defined locations. The last double column of Tab. 2 reports results for averaging all the kernels computed for *bbox*, *up-bbox*, *low-bbox*, and *ex-bbox*. PA still outperforms DF on many classes, suggesting the non-diminishing advantage even for multiple kernel combination.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[3] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 4.0.1. http://people.cs.uchicago.edu/ rbg/, 2012.

[4] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE PAMI*, 2010.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[6] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[8] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.